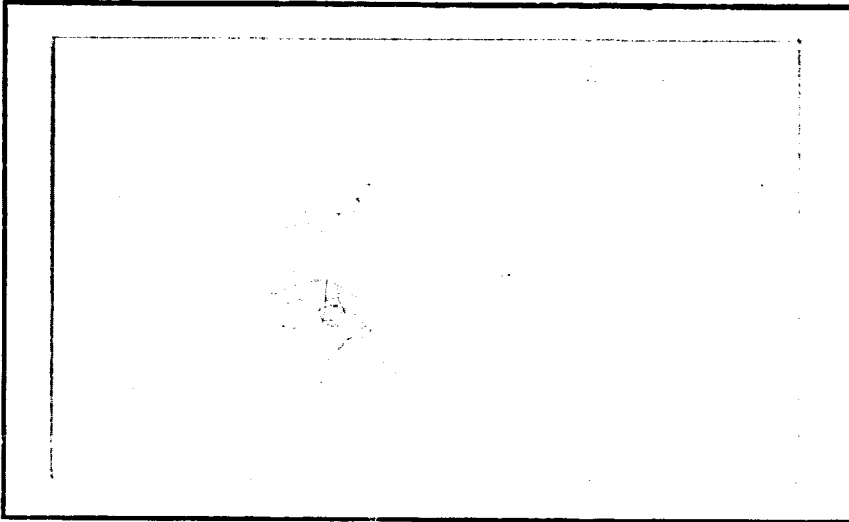


## **General Disclaimer**

### **One or more of the Following Statements may affect this Document**

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

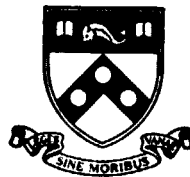
CR-171 897  
c.1



{NASA-CR-171897) MOTION ANALYSIS REPORT  
(Pennsylvania Univ.) 33 p HC A03/MF A01  
CSCL 12B

N86-10896

Unclas  
G3/66 27492



**UNIVERSITY of PENNSYLVANIA**  
*Department of Computer and Information Science*  
*The Moore School of Electrical Engineering*  
**PHILADELPHIA, PENNSYLVANIA 19104**

# **Motion Analysis Report**

**May 31, 1985**

**NASA Contract No. NAS9-10034**

**Norman I. Badler,  
Principal Investigator**

**Department of Computer and Information Science  
Moore School D2  
University of Pennsylvania  
Philadelphia, PA 19104**

# **Motion Analysis**

## **Table of Contents**

<b>1. Introduction</b>	<b>1</b>
<b>2. Image Sequence Analysis</b>	<b>3</b>
<b>2.1. Motion Data Acquisition</b>	<b>3</b>
2.1.1. Image Data Acquisition by Manual Methods	5
2.1.2. Image Data Acquisition by Automatic Methods	5
2.1.3. Image Data Acquisition by Semi-Automatic Interactive Methods	7
2.1.4. Non-Image Motion Data Acquisition	8
2.1.5. Summary of Techniques	8
<b>2.2. Converting Two-Dimensional Data to Three-Dimensions</b>	<b>10</b>
<b>3. Jointed Motion Analysis</b>	<b>12</b>
3.1. Model-Driven Analysis	13
3.2. Kinematics and Dynamics	14
3.3. Motion Data Models	16
<b>4. Motion Data Display and Description</b>	<b>17</b>
4.1. Graphical Display	17
4.2. Real-Time Motion Playback	18
4.2.1. Current Implementation	21
4.2.2. Program Prep	21
4.2.3. View	22
4.2.4. Necessary Extensions	23
4.3. Textual Descriptions	23
<b>5. Conclusions</b>	<b>24</b>
<b>6. Schedule and Resources</b>	<b>25</b>
<b>7. Bibliography</b>	<b>27</b>

## **Motion Analysis**

### **List of Tables**

<b>Table 2-1:</b>	<b>Motion Analysis Method Comparison</b>	<b>9</b>
<b>Table 6-1:</b>	<b>Motion Analysis Schedule</b>	<b>25</b>
<b>Table 6-2:</b>	<b>Motion Analysis Resources</b>	<b>26</b>

## 1. Introduction

*Human motion analysis* is the task of converting actual human movements into computer readable data. Such movement information may be obtained through *active* or *passive* sensing methods. Active methods include physical measuring devices such as goniometers on joints of the body, force plates, and manually operated sensors such as a *Cyber* dynamometer. Passive sensing de-couples the position measuring device from actual human contact. Passive sensors include *Selspot* scanning systems (since there is no mechanical connection between the subject's attached LEDs and the infrared sensing cameras), sonic (spark-based) three-dimensional digitizers, *Polhemus* six-dimensional tracking systems, and image processing systems based on multiple views and photogrammetric calculations.

In a zero-gravity environment, some of these sensing systems become cumbersome or even impractical. We will not extensively review systems already in place in the AML at JSC, but will rather concentrate on sensor systems that are either unavailable at JSC at present or which hold the most promise for effective zero-gravity motion analysis.

In particular, the *Cyber* and *Selspot* systems will not be discussed in detail. We simply note that there would be no apparent problems using a *Cyber* machine in an appropriately configured zero-gravity environment if suitable restraints were available. On the other hand, using a *Selspot* system in zero-gravity would create novel problems in the area of sensing significant translational motion of a body since the sensing cameras must remain fixed and their views unobstructed. This may be possible in large spaces (such as were built into SKYLAB) but which are not found in the Orbiter. An additional problem is the encumbrance of the subject with the LED umbilical cable which may unwittingly affect motion performance in zero-gravity.

One solution to the umbilical cable is to use a video system and special hardware designed to track specially colored dots on a moving subject in real-time [15]. Unfortunately, the resolution of this system is too coarse for detailed human motion analysis as the dots must be rather large in order for them to be properly sensed by the video camera. There has been no known follow-on to this system which would actually provide digital locations of the dots suitable for further analysis. The device has been used only for entertainment purposes.

*Kin-Com* is a recent device to rival the *Cyber* system for active force and motion

## Motion Analysis

sensing [26]. This device offers a turnkey computer graphics plotting capability and the ability to operate in three data collection modes: isometric, isotonic, and isokinetic. Further information on this system does not appear to be essential since its functionality overlaps the Cybex and host computer set-up presently installed at JSC.

Another passive motion sensing technology available today is the six-dimensional tracking system offered by Polhemus Navigation Sciences. One such device was obtained on the current NASA contract so we have first-hand experience with it. It is worth pointing out that the cost of the electromagnetic technology and digital computation which drives the process has been reduced significantly in just two years. The present cost of the most accurate system is about \$20,000 (the system we have purchased for NASA) but a recently announced *lower* resolution but much more *portable* version sells for \$2,995. The accuracy of the former is on the order of 1 mm, that of the latter, about 6 mm. The sensor is a small plastic cube and must be attached through an umbilical cord to the small computer housing. The electromagnetic source is small and portable and can be mounted in any non-metallic area. Unfortunately, the combination of low resolution and limited spatial coverage (about a cubic meter) make this device less attractive as a motion analysis instrument. Its primary application will probably be as an inexpensive pointing device for direct computer input (such as menu picking, drawing, and object positioning) where absolute accuracy is not too important, but relative control is.

A sonic digitizing pen enables the direct sensing of the three-dimensional position of a spark generated at the tip of a stylus in approximately a cubic meter space. The sound of the spark is picked up by strip microphones. From the position and timing of the sound, the position of the tip may be determined. Since the technology relies on sound waves, it is very sensitive to occlusion (hiding) of the waves by objects in the active sensing area. The spatial resolution is to within a few millimeters. The high frequency spark may not be desirable in a zero-gravity environment with much sensitive electronic equipment in the area. We therefore feel that this device is not appropriate to the motion analysis task.

There is really only one alternative to true passive motion sensing where the environment and the subject need not be specially prepared and, moreover, motion information is not necessarily extracted in real-time or on-line. That alternative is computer image analysis. The input is a sequence of film or video images of people in

## **Motion Analysis**

motion, usually taken some time prior to analysis. The subjects are unencumbered and require no special attachments. The image data may be collected with the known intention of doing motion analysis or may be processed after the fact. The former is frequently done in sport analysis and training to assess forces or weight distribution; the latter may be done with existing imagery such as the extensive SKYLAB visuals to determine work patterns and locomotion in zero-gravity.

In the remainder of this report we will examine the parameters and potential of this process in detail. The primary research papers in this survey are presented in the Bibliography.

### **2. Image Sequence Analysis**

The task of human motion analysis from image sequences includes the following subtasks:

1. Observe, either with manual methods employing a human operator or with automatic (algorithmic) methods executed by a computer, a sequence of images of a human and extract the two-dimensional projections of various body features onto the image plane.
2. Take the two-dimensional coordinates from (1) of, say, body joints or other distinctive features, and compute or infer the three-dimensional coordinates of those features.
3. Find the paths of the features from (2) and compute required joint angles, velocities, accelerations, forces, torques, *etc.*
4. Display the results of motion analyses from (3) using computer generated graphics.
5. Use the parameters from (4) to derive models of human motion in the tasks observed in (1), for example, to develop isotonic strength models, view explicit reachable spaces, or study methods of human body locomotion in zero-gravity.

We will look at each of these steps in turn.

#### **2.1. Motion Data Acquisition**

The first goal of a motion data acquisition system is to observe, either with manual methods employing a human operator or with automatic (algorithmic) methods executed by a computer, a sequence of images of a human and extract the two-dimensional projections of various body features onto the image plane. This task is regularly and effortlessly performed by people and their visual systems, yet the automation of the



## **Motion Analysis**

visual process remains a persistent and rather elusive challenge to computer scientists.

The image features of interest may include:

- body extremities (hand, foot, head)
- fixed points (eyes, nose, mouth)
- joint centers (elbow, knee, ankle, wrist)
- implicit joint centers (hip, clavicle, head-neck joint)
- implicit spinal curvature
- surface orientation (pronation or supination of forearm or lower leg)
- support or restraint (contact with environment)
- body segment size and shape (muscle contractions, breathing, anthropometric measurements)

While a more complete definition of each of these features in terms of image data is warranted, we will leave the concepts at a reasonably clear intuitive level. The characteristics of these features in a digitized image, for example, will be left to Section 2.1.2 where the information is much more essential.

In order to approach the motion data acquisition problem in a fashion meaningful to the general tasks expected under this contract, we will divide the methods into four groups:

1. Image data acquisition by manual methods
2. Image data acquisition by automatic methods
3. Image data acquisition by semi-automatic interactive methods
4. Non-image motion data acquisition

The reason for this decomposition is that the different methods tend to be the provenance of dissimilar research groups, but our goal is to tie the different pieces together into a coherent discussion of the human motion analysis problem. We include a discussion of non-image data acquisition methods both for parallelism in presentation and also to emphasize that the methods are not mutually exclusive.

## Motion Analysis

### 2.1.1. Image Data Acquisition by Manual Methods

The most prevalent method for obtaining human motion data involves taking image sequences on film, video, or high speed photography, projecting the sequence one frame at a time onto a screen or transparent digitizing tablet, and manually selecting and digitizing the features of interest [25, 47, 6, 41]. Bioengineering and sports medicine researchers are notable users of these methods: they are inexpensive in terms of computers and peripherals, relatively effective in producing motion data, involve rather minimal programming, and require little specialized training for use.

At its crudest, the image is projected onto a grid of lines from which the coordinates are read and later keyed into a computer. More efficient is the use of the digitizing tablet to perform the conversion to two-dimensional coordinates directly on the image. While conveniently interactive for the user, the process tends to be very tedious and error-prone due to the manual selection of points to digitize.

With point digitization schemes there is usually no provision for analysis of the image other than the location of joint or fixed features. Indication of orientation, curvature, or even segment shape is normally difficult and is avoided.

The data obtained may depend on the spatial resolution of the sensor and usually requires several post-acquisition steps to render it safely usable, including filtering to smooth the quantization and positioning errors and conversion to the three-dimensional positions of the real body. We will address these issues later in Sections 2.2 and 3.3.

### 2.1.2. Image Data Acquisition by Automatic Methods

Efforts by several computer vision researchers have been directed toward the automatic analysis of motion [33]. In this method, motion picture or video images of object and possibly observer movements are presented to the computer as a sequence of digitized frames. The images are scanned by algorithms to extract the interesting features, such as edges, and perform various operations on these features, such as change detection, edge connection, region finding, centroid computation, and so on [9, 56, 1]. The process is highly dependent on the quality of the images, the lighting conditions, the contrast of the objects against the background, the size and shapes of the moving objects, the spatial resolution of the images, and the temporal resolution (time interval between frames) of the sequence.

The efforts in the literature which describe attempts at *automatic human motion*

## Motion Analysis

*analysis* are minimal. The bottom line on automatic methods is that they are still in their infancy and do not offer particularly short-term hopes of providing robust and effective motion analysis. Most of the methods presume some lower level image analysis steps that provide two-dimensional features for motion analysis.

Only O'Rourke and Badler [37] have reported on a human motion image analyzer that attempts to perform low level image feature determination as an integral part of the system. But even in their system, the images were assumed to be quite well-formed, having in fact been computer graphically generated as gray-scale images by the *BUBBLEMAN* body display program. The automatic analysis depends on a model of the human figure which drives the image analysis component to search for extremities in likely regions based on the body's ongoing motions. The program's image search for features was goal-directed and therefore rather efficient: since in general the program knew approximately where the body extremities should be found (as long as motions were not too fast), only about 10% of the image pixels had to be examined to determine the new, next location of the feature.

The search for body extremities turns out to be fortuitous in the sense that not only are these the most reasonable regions to search for due to the high curvature exhibited in the image at the limb ends, but also are most informative in terms of motion information [43]. In an independent study, Tartter and Knowlton found that much of the information in an American Sign Language "utterance" could be conveyed by simply transmitting the locations of 27 moving dots on the wrist, hand, and fingertips [49]. The resulting video "moving dot displays," were interpretable to American Sign Language readers watching a video display. There were no digital conversions performed here, only the process of suitably thresholding a high contrast video image in real-time.

O'Rourke and Badler's image matching process could be told or could "learn" the joint-to-joint lengths of body segments, and is therefore amenable to image analysis of either known or unknown figures. Since it uses a model of the human body coupled with a positioning simulator, it is able to deal with occlusion (hiding) of body parts and three-dimensional motions in a rather general way. The key concept used in forming the motion model is a *constraint network* in which information on segment length restrictions, joint angle constraints, balance, and support requirements may be embedded and used to drive the analysis process.

## Motion Analysis

### 2.1.3. Image Data Acquisition by Semi-Automatic Interactive Methods

Midway between the manual and automatic methods of image analysis are techniques pioneered mainly by computer graphics researchers. The most important of these involve the use of computer assisted film or video projection with flexible "housekeeping" features to streamline and enhance motion data acquisition. The first system of this sort was *Galatea*, built by Futrelle, Potel, and Sayre at the University of Chicago [44]. *Galatea* is still in production use, solving a variety of difficult motion analysis tasks including, but not at all limited to, human motion from one or two views [30]. The *Galatea* user views overlapped (projected) motion imagery and computer graphics and can draw or select points on the common screen. The motion data may be played forward or backward at any desired speed, including stopped single frames. The authors note that the ability to vary the speed and have the selected points replayed in synchrony with the data permits the user to digitize motions that are not apparent in any single image. The user's own perceptual mechanisms act as a control to insure accurate digitization. This is especially important for moving images since they tend to be blurred by the finite shutter of the movie camera or the sampling period of the video camera. This blur in each single frame is a significant source of data digitization error in fast motions. The housekeeping features of *Galatea* include keeping time ordered lists of data point positions associated with a feature, displaying information such as clocks and point paths, and allowing the user to edit the feature point selections until the results are satisfactory.

A few years ago (around 1981), Badler tried to combine the strengths of the automatic approach, namely constraint propagation, with the strengths of the semi-automatic approach by using human vision for the image analysis and data verification step. The effort failed solely from the lack of adequate programming support and equipment rather than any conceptual fault. This direction is worth pursuing once again.

It would be our recommendation to NASA that motion analyses be performed with a hybrid system of this sort as a means of optimizing data acquisition efficiency and accuracy against software development costs and complexity. The hardware technology to build such a motion analysis system is available and engineering interactive graphics systems is a reasonably well-understood problem. The new generation of fast raster graphics systems which allow the mixing of video and synthetic imagery is ideal for this application.

## Motion Analysis

### 2.1.4. Non-Image Motion Data Acquisition

There are a number of ways to collect motion data which do not rely on image analysis and are therefore apt to be more effective in providing accessible, accurate data. These methods include systems such as the Selspot infrared sensing system, the Polhemus six-dimensional digitizer, the three-dimensional sonic "pen," and directly attached joint motion sensors such as goniometers [11]. The data supplied by these sensors should be integratable into a more general motion analysis system. Indeed, several groups(e.g. [11, 18]) use such inputs as the basis for realistic motion display, though the raw data requires some filtering before it can be used.

There are parallels between the visual and non-visual media in terms of the kind of data collected, though clearly some types of motion features are easier to detect with one than the other. For example, the Polhemus is excellent at sensing orientation, while image analysis is not. Also, the Selspot system provides a fixed association between a body point and its infrared LED while the body moves in space. This association is more difficult for image analysis which must continually re-establish visual feature locations (and endure the consequent errors) from frame to frame. The most advantageous situation would be where such non-image data augments any visual data to mutually support and improve the accuracy of the analysis. For example, were Selspot data provided to a Galatea-like system, one could immediately view the combination of LED and visual data to determine if the data suffers from unwanted occlusion effects. Combining the non-image data with the known body segment length constraints might enable the accurate determination of joint centers and the simple verification of correct motion tracking.

Since much of the available imagery requiring motion analysis (such as the SKYLAB films) is not corroborated by non-image sensory data, we will assume that this approach is not as crucial to the expected tasks for a motion analysis system at this time. Such a system would find maximum usefulness in future efforts to study human movement in low or zero-gravity environments or to build a general body strength atlas for arbitrary or particular movements.

### 2.1.5. Summary of Techniques

Table 2-1 below summarizes the assets and liabilities of each of the four data acquisition methods. The entries are based on subjective criteria and are meant more for relative comparisons.

## Motion Analysis

**Table 2-1: Motion Analysis Method Comparison**

METHOD	Equipment cost  (excl. computer)	Speed of  data acq.	Computer  prog.	Operator  training	Accuracy	Feasibility
Manual	modest (~\$5K)	slow	simple	little	low	available
Automatic	modest (~\$6K)	medium?	hard	none	medium?	research
Semi-Auto	high (~\$80K?)	medium	moderate	some	high?	prototypes
Non-Image	medium (~\$20K)	real-time	simple	some	good	available

The *equipment costs* are based on estimates of the additional hardware required to construct a single digitizing workstation. For manual methods, the cost represents a movie or video projector and a digitizing tablet. For automatic methods, the cost amounts to a high resolution video digitizing camera. One should note, however, that commercial systems for image sequence analysis run into the \$100K range primarily because of the vast (disk) storage requirements for digitized imagery. This cost has not been factored into the chart. For semi-automatic methods a video digitizing camera is needed plus an interactive graphics display with at least a modest real-time playback capability. The latter could be any of several high-performance raster graphics systems or workstations of the Silicon Graphics IRIS class. (Although it is not noted in the table, an automatic system should probably have the same real-time playback capabilities of the semi-automatic system for data validation. In that case, the automatic system is roughly the sum of the semi-automatic method and the special image digitizing hardware.) Finally the cost of the non-image systems is based on the approximate cost of a Selspot, Polhemus, or Kin-Com system..

The *speed of data acquisition* is based on estimates of the relative time to digitize all of the subject's body joints in one frame of motion imagery. The time for the automatic methods is a guess since no satisfactory operating prototype exists. In general, current image analysis systems are not significantly faster than trained human operators at complex recognition tasks (and may not perform nearly as well). The principal strength of the semi-automatic approach is to trade-off human response time for the improved intelligence in performing the image analysis task. The result should be faster than purely manual techniques. Finally, the non-image methods by definition acquire data in real-time.

## **Motion Analysis**

The computer programming required to implement each of these methods varies considerably. The automatic method is not just hard, however, but perhaps not known!

Operator training is self-explanatory. The manual and semi-automatic methods require the operator to interact with the computer in some fashion. For the semi-automatic system the interaction is coupled with a real-time playback and data adjustment procedure which complicates the process somewhat. That interface may be nicely engineered, however, as the Galatea system demonstrates.

The accuracy of the data collected in manual methods depends on the projected image resolution and the hand and eye accuracy of the operator. We have noted the typical lack of control on the body segment lengths which should constrain point input. Automatic methods depend on the digitizer resolution and, since that is frequently not very good, on sub-pixel feature extraction (see the next section). These methods use gray level image information to try to locate feature centers assuming that they do not lie exactly on grid positions. The accuracy here depends on the feature extraction algorithm and the intensity distribution of the data. Semi-automatic methods can combine operator inputs with the body segment model to accurately locate points. The accuracy of non-image methods appears sufficient, but noise reduction and filtering are often necessary as noted above.

Finally, the feasibility of manual and non-image methods are proven, while automatic and semi-automatic methods require further research and development. Of these two, the semi-automatic methods offer the best medium-term prospects because we can maximize performance by utilizing human visual intelligence for time-consuming sequential image analysis algorithms. Interactive computer graphics systems with real-time playback and record keeping augment human pattern recognition at the cost of (potentially) lower, but more accurate, data throughput. There is, however, no absolute basis of comparison at this time.

### **3.2. Converting Two-Dimensional Data to Three-Dimensions**

Once two-dimensional feature points have been determined in the image, those points must be converted to three-dimensional information. There are generally two approaches to this problem: the first assumes enough control over the situation that multiple views (either with mirrors or multiple cameras, such as stereo) may be taken; the second works solely from one view. In the latter case, the fact that there is a

## Motion Analysis

sequence of images is often of crucial importance.

The problem of reconstructing three-dimensional points from pairs of two-dimensional points in stereo views is a much researched and reasonably well-understood topic [16, 46, 6, 33, 31, 52, 34, 57, 2, 59, 35]. With its basis in simple trigonometry, it forms the essential component of photogrammetry and biostereometrics. There are two problems in the reconstruction; they are not in the mathematics of the conversion but rather in the initial association of corresponding points in the two (stereo) images and the resulting coordinate values of those points. Manual methods of establishing point correspondences are frequently employed, since a point clearly visible in one view may be occluded in the other. It may not be clear which points correspond to which: for example, consider the perception of a smooth surface such as a sphere. Correspondences to establish depth of the surface from the observer may be impossible to obtain. Clearly other psychological and perceptual cues are necessary. Recently major efforts have been made in the Artificial Intelligence community to understand stereo vision and build automatic methods for performing stereo correlation over a simple image [31, 20]. The lack of a sufficiently dense set of distinguishable feature points over the surface of a human body presents a barrier to the straightforward application of these automatic methods.

The second problem is that the depth reconstruction computation is quite sensitive to errors in feature location. The depth computation becomes more imprecise as the point under consideration becomes more distant from the camera baseline. Given that the image feature points are often only located to image resolution (pixel) accuracy, the inherent error in the reconstruction is considerable. To counteract this effect, the location of an image feature may be determined to *sub-pixel* accuracy by attempting to find its position from image intensity analysis of the pixels in the surrounding region [17]. Unfortunately this method is only effective when the feature point geometry is known, for example, to be a point or a disk.

When stereo reconstruction is not possible, *moire interferometry* may be employed. In this technique, thin parallel stripes of light are projected from two sources onto a body or body surface. The resulting band patterns may be analyzed to determine the depth of each part of the band from the viewing camera, thus determining the three-dimensional position of many points on the surface. Moire interferometry has been successfully applied to change detection (difference between two images), though not to



## Motion Analysis

general motion analysis.

The problem of determining the correspondence between feature points in stereo is similar to the problem of determining the correspondence between moving points in two views. In the former case the camera model is assumed known, while in the latter case the motion of the joints involved are typically unknown. Thus we should expect that the point correspondence problem for moving images is harder than stereo, and in fact this is the case. Determining moving point correspondences algorithmically is motivated by the apparent ease with which human observers recognize other human figures and their complex motions (walking, dancing) even when only presented with a series of moving dot displays [24, 14, 45]. The correspondence method can be as simple as "nearest neighbor" in the next image (which does not work too well unless the motions are very slight [45]), or can depend on more global properties of a neighborhood of moving points [52, 53, 37]. Quite recently, Jenkin and Tsotsos [22] have had interesting results tracking human figure moving dot data by simultaneously using motion and stereo cues.

Significantly, all the automatic correspondence methods depend on *point* correlations and are therefore relatively unsuitable for more general image feature matching. The problem of matching non-point features, such as edges or regions, from frame to frame is more difficult [56, 53]. To our knowledge no one has successfully implemented methods to automatically correlate stereo views of arbitrary human motion from real, unprepared image sequences.

### 3. Jointed Motion Analysis

The three-dimensional information extracted from moving features of the image sequence must be segmented into individual motions for each joint and body segment, then validated against a model of allowable human movement. This stage is crucial to the proper interpretation of the motion paths of the body joints, since those joints that are not easily located (the "indirect" joint centers) must have been visually inferred from surrounding features. For example, motion of the elbow or knee is not too difficult to track from frame to frame unless the limb is completely extended; the precise location of the joint center may be hard to find in those frames. The problem is worse for deeply centered joints such as the hip or the head-neck joint. A joint motion model must relate data obtained from the image sequence to the known model of human structure.

The first step in jointed motion analysis is segmenting motions of the body as a

## Motion Analysis

whole from whatever background is present. In controlled environments (*i.e.* where the background is uniform and neutral, or where applied dots or lights may be tracked) separation of figure from ground is not difficult. In arbitrary environments, however, the separation may be quite error-prone or even impossible (*e.g.* consider camouflage). The image processing step has been examined by many researchers (*e.g.* [1, 32, 38]).

The second step in jointed motion analysis is finding the individual segment motions. This process involves organizing the numerous pieces of actual motion data into coherent motion paths for each body joint. First, global body translation and possibly rotation are determined if possible. When done, the body coordinate system becomes a fixed frame of reference for more distal motions of the limbs. Badler [6] determined the body coordinate system manually by tracking the center hip (roughly the body center of gravity). O'Rourke and Badler [37] established the body center indirectly from the limb positions and constraint propagation to determine valid torso positions. Neither of these approaches achieved the direct specification of body (torso) orientation. Tsuji and Asada [51, 4, 5] use pattern recognition and clustering techniques to isolate motion displacements of articulated segments undergoing simultaneous rotations.

### 3.1. Model-Driven Analysis

When the subject is known, computer based image analysis methods can be tailored to the application to improve efficiency as well as performance. A number of researchers have investigated *model-driven* vision where the control and search tasks in the image or image sequence are guided by a stored prototype. This does not finesse the problem at all; the system must still locate instances of the prototype in arbitrary image locations, with three-dimensional orientations, and possibly in spite of occlusion. While this approach is attractive but not crucial for certain computer vision problems, it appears essential for human motion analysis as evidenced by the publications [37, 58, 3, 53].

The structure of the human body skeleton is, to a first approximation, a collection of linear segments connected at joints. Thus, there are no real extension or contraction motions possible. The motions at a joint are known to depart from strict spherical or revolute rotations, but those variations may be modeled more exactly if we are prepared to make our models a function of joint angle position. We view this as a desirable, though longer term goal, and will ignore it for now.

## Motion Analysis

Maintaining the structure of the body requires that joint displacements in three-dimensions be resolved against the fixed segment lengths and joint constraints of the skeleton. When such conditions are not respected, the resulting raw joint data may show the body segments changing lengths during a movement. Besides being incorrect, such anomalies may indicate poor data collection or errors in the conversion from two dimensions to three.

There have been several attempts to use the body structure in performing motion analyses; rather than treating the figure as just a collection of moving points, the relationships between the points (joints) are exploited. O'Rourke and Badler [36, 37] used the *constraint propagation* method to insure that no joint was positioned outside of the area it could actually occupy given the locations of its adjacent joints. When this condition is successively propagated to all body joints, the result is a position which is compatible with all data as well as the body skeleton. Webb and Aggarwal [55], and recently, Chen and Lee [12, 29] have also developed methods for relating image data to admissible configurations of body joints. It is our belief that the work by O'Rourke and Badler is a more general scheme for joint motion modeling because Webb and Aggarwal require a so-called "fixed axis" assumption for joint motion (no limb may be rotating around a rotating proximal segment), while Chen and Lee only provide a way to reduce the possible configurations down to a manageable set by invoking strong assumptions of a walking posture.

The conclusion to be reached for human body motion analysis is that an effective *model* of human body geometry and motion must be used *during* the analysis and not just after geometric decisions have been made. The model thus requires up to three degrees of freedom for each body joint, not to mention the curvature of the spine. There are few, if any, techniques available for translating the enormous number of configurational possibilities of the human body into valid configurations of the model. We shall explore some of the techniques in the next section.

### 3.2. Kinematics and Dynamics

Human joint motion models may be based on the kinematics of a linked structure. Kinematic solutions are broken into two classes: *forward kinematics* and *inverse kinematics*. Forward kinematics is concerned with finding the positions of points on a body when all the joint angles between body segments are defined. This is the same as mapping the angular joint coordinates to Cartesian (spatial) coordinates. The forward

## Motion Analysis

kinematic problem is always solvable by multiplying homogeneous transformation matrices. TEMPUS performs forward kinematics during graphical display of a human figure. Inverse kinematics is the mapping of Cartesian coordinates back into angular joint coordinates. That is, given a point which a body wants to reach, the inverse kinematics would solve for all the joint angles. The inverse kinematics problem cannot, in general, be solved analytically.

Forward dynamics of the human body is concerned with determining the forces that a body can apply for a required path given its initial conditions (position, velocity, and acceleration). The dynamic problem can also be inverted. When an external force is applied on a body the position, velocity, and acceleration can be determined as a function of time. The external force can also be a function of time. Solving for the various parameters requires the integration of the equations of motion. There are also two equivalent methods to the direct solution of the equations of motion which may yield computationally more efficient solutions: Lagrange equations and Newton-Euler equations.

The robotics literature (*e.g.* [42]) is full of methods for computing the joint angles of a manipulator given the Cartesian (three-dimensional) location and orientation of the end effector (hand). The reach positioning algorithm of Korein [28] provides a way of computing the elbow and shoulder (knee and hip) angles given the location of the fingertip, palm center, or wrist (toe, foot center, or ankle). With such a computational model, only the location of the three-dimensional end effector need be gathered from the image sequence; the other limb angles may be computed. The use of robotics-type algorithms has recently been extended by Girard and Maciejewski of Ohio State University [19]. Their simulation of a multi-legged walking vehicle depends in a Jacobian matrix formulation of joint kinematics and a pseudo-inverse solution which determines joint angles given the position and orientation of the end-effector.

The limitation to these methods is the difficulty in extending them beyond the two or three link chains of the human limb and accommodating arbitrary restraints. This difficulty has limited previous systems (such as *Combiman* [8] and *Sammie* [27]) which attempted human reach or motion modeling to restricted activity domains: principally seated figures with lap or shoulder restraints. In zero-gravity, the restraints are more general or even non-existent. Therefore a structure is needed which is able to accommodate fully articulated motion.

## Motion Analysis

Given an arbitrary set of position and orientation constraints we must solve for the resulting configuration of all the joints. General inverse kinematics for the entire body structure lead in two directions: one is toward the explicit representation of constraints as in O'Rourke and Badler's system, the other is toward general mechanism solvers as exemplified by mechanical engineering analysis programs. Constraint propagation has unsolved problems relating the orientation dimensions of the kinematic process. The resulting six-dimensional spaces are very unwieldy for simple geometric operations. Moreover, the costs (in time) for manipulating such high dimensional spaces probably renders their use impractical.

General mechanism solvers exist for open or closed loop systems. In closed loop systems there is more than one linkage pathway between some pair of points; in an open loop system there is no such path. A human figure in free-fall and not in contact with itself is an open loop system. A figure standing on two feet or grasping restraints with both feet and hands has closed loop components. Clearly closed loop models are required for TEMPUS.

The general mechanism solvers are represented by several computational systems. Most of these are either expensive (being commercial products), particular to one computer, or inefficient (incorporating solution algorithms which are inherently expensive). The first two categories are exemplified by ADAMS [39]; the third by IMP [23]. We have one system, DYSPAM [40], available on our VAX which we are evaluating for possible incorporation into the TEMPUS body model. DYSPAM can solve three-dimensional (spatial) mechanisms by inverse kinematics and dynamics by the Lagrange equation method. The formulation of kinematic problems leads to a system of nonlinear algebraic equations. The equations are then solved by applying the Newton-Raphson procedure.

### 3.3. Motion Data Models

Given that the three dimensional positions of body joints have been determined over some time interval, it is likely that some quantities will be computed from the data such as velocity, acceleration, force (if masses are known), and torque (if moments of inertia are known). Since these depend on derivatives of the initial displacement data, errors (even of the apparently benign sort, such as the resolution of the digitizing device) are greatly exaggerated.

## Motion Analysis

Consequently, raw joint motion data is typically smoothed by curve fitting techniques to insure that derivatives will be a reasonable reflection of the true motion model. In some applications, Fourier analysis followed by filtering to remove the high frequency noise is used to do the motion smoothing, but this technique is expensive and most suited to cyclic (repetitive) motions such as gait (walking, running, etc.). We would propose a filtering method based on B-spline curve interpolations. These curves offer considerable control over the fitting process and, because they are piecewise polynomials, are easy to differentiate.

Steketee and Badler [48] have been investigating motion models primarily from a generative point of view. Their approach is to describe the motion of any parameter by a pair of interpolatory B-splines. One describes the relationship between the data collection ("keyframe") times and the positions measured, the other is used to adjust the data collection times to vary motion kinetics. Such a technique may be used to adjust the kinetics (accelerations) and the data points *independently*. Given a real-time playback capability, therefore, the effectiveness and correctness of the analysis may be assessed by visually superimposing the analysis over the original data. This idea originated with Galatea and could be applied to the interactive motion analysis envisioned.

### 4. Motion Data Display and Description

Motion data may be displayed graphically or described with text. We will examine possibilities for both in turn.

#### 4.1. Graphical Display

There are many ways to display motion data for the human figure, the most notable of which show the figure itself in motion. A number of possible techniques of value are listed:

- Display graphs of displacement, velocity, acceleration, force, and torque versus time.
- Animate the human figure in real-time. This is usually done with a stick figure for adequate speed, but with high performance display devices such as the IMI 500 vector graphics display, thousands of lines may be manipulated in real-time.
- Slow down or speed up motion data, especially in the context of the human figure display cited above.

## Motion Analysis

- Color code the joints or segments of the body according to the value of some parameter of interest, such as maximum velocity, acceleration, torque, force, joint angle, etc.
- Display other computed quantities derived from the motion data such as the center of gravity, points of support or restraint, angular velocity or momentum. The latter especially lends itself to creative display techniques, such as dynamically changing vectors and interactive application of user-supplied forces to test the model's performance.

The approaches most likely to be necessary for motion data analysis are real-time playback and the display of motion variables versus time. With these methods the accuracy of motion data obtained from any of the analysis techniques mentioned above may be validated or adjusted to remove spurious or erroneous data points. The other techniques offer the user a global view of motion data over the body structure and over time by displaying of a very large number of motion parameters (on a per joint or segment basis, for example), while showing their evolution in time. This type of display may have to be created frame-by-frame and played back in real-time. It is possible, however, that the color display monitor of the IMI 500 could perform this task directly.

We feel that the IMI 500 display purchased on the current contract is suitable for motion playback, but cannot simultaneously display the source images. The solution to this specific problem entails optically overlapping the original image sequence and the graphics. This is a bit tricky (due to registration problems between the two images [44]), so recently some of the Galatea developers have used a color raster display where the video source images can be electronically merged with the synthesized graphics. This is the direction this project should take since the expected amount of motion data should be comfortably displayable on new generation graphics workstations.

### 4.2. Real-Time Motion Playback

This section describes the design and implementation of a real-time graphics animation playback system. Our initial specifications called for two programs to be written. One program on the VAX would accept a file containing a series of scene descriptions (such as those used by Frank Crow's rendering system [13]; see [7]), process these scenes into an animation file, and transport the animation file over to the IMI 500 graphics workstation. The other program on the IMI would play the animation back in "real-time", allowing some interactive user control of the speed. For reasons discussed later, the initial implementation of the system runs on the IRIS 1400 graphics

## Motion Analysis

workstation.

The program on the VAX takes a series of scene descriptions and composes an animation file containing an initial scene description and a series of changes in the scene which are supposed to take place from one frame to the next. The initial scene description contains all of the commands necessary to change the null scene into the image for the first frame. The format chosen for the animation file must leave as little processing as possible up to the graphics workstation, yet it should also be general enough to be used by several devices and completely describe all possible changes that may take place in a scene. Specifically it must be able to effect each of the following changes.

1. Changes in the structure of the object hierarchy, attachment, detachment, etc.
2. Changes in the relative transformations contained in the object hierarchy.
3. Changes in the object description table, additions, deletions and modifications.
4. Changes in the viewing parameters.

Each of these commands must be in a form which explicitly indicates the operation to be performed as well as the objects involved.

The program on the IMI should be broken up into two parts: a frame manipulator and a display program. The frame manipulator would have several responsibilities. It should handle the input and storage of frames, the execution of the commands contained in the animation files, and the adjustment of the playback rate as indicated by the user. It should execute animation file commands by creating and changing an object hierarchy which represents the current status of the scene.

The display program should be an independent program running on the IMI's graphics processor. Once started it should run continuously until the animation is halted. Its responsibility should be to maintain the image on the screen by traversing the object hierarchy created by the frame manipulator.

Another important consideration is to create a system which will be readily accessible and easy to modify. This requires that all software be as machine independent as possible and that the file formats and command structures be applicable to any hardware configuration on which the system may be implemented.



## Motion Analysis

During the implementation of the prototype system, several changes were made to the original design. Of these changes, two decisions in particular had a large impact on the outcome of the project: to use the IRIS 1400 instead of the IMI 500 as the graphics workstation, and not to represent object hierarchies in the display software. Both of these decisions were made for expediency and must be reversed in the next cycle of implementation.

The decision to use the IRIS instead of the IMI for now does not represent a change in commitment from one machine to the other. It is still the intention, as it was all along, to have the playback system implemented on both machines. The original decision to use the IMI was made simply because the IMI was the only machine available and because it is also an unquestionably more powerful device. Software development, however, is easier on the IRIS as it has a friendlier programming environment. This change affected only the machine-specific display software created for the current system. The same file formats and scene analyzer can be used for both the IMI and the IRIS.

The second decision, not to represent object hierarchies on the display device, was taken simply due to display speed. Relieving the workstation of the responsibility of maintaining object hierarchies allowed for the implementation of a much simpler animation file format, and also greatly reduced the amount of work required to process one frame in the display program.

Without the implementation of object hierarchies, the animation file format only needs to handle three types of commands: object descriptions, object calls, and viewing parameter settings. Object descriptions use the same format as Crow's system. Object calls consist of a pointer to an object description and an absolute transformation matrix. Each frame then consists of a list of object calls for all objects active in that frame. Each call places that particular object in its proper position for that particular frame. The viewing parameters are the same as those used by Crow's system. Their values are passed through directly to the display device and accounted for using the graphics workstation software. This is done because the actual transformations performed for the view settings are often machine specific.

This organization improves the system's performance by cutting down on the amount of processing done after the initial scene analysis. Initialization now only

## Motion Analysis

requires object descriptions, instead of complete object hierarchies, and the display program now has the absolute matrices provided for it instead of having to derive them itself by traversing the object hierarchy.

There are drawbacks to not maintaining object hierarchies on the display device, although they are not as significant as first thought. Without an object hierarchy the workstation cannot independently manipulate object positions and relationships. This capability is essential if the playback system is to be used as a full motion control and animation editor. It would be a straightforward matter to read in the object hierarchy from the VAX when object positioning is required. This interruption would be noticeable, yet would probably not be too much of a bother as the user would be anticipating a change in the functionality of the program anyway.

### 4.2.1. Current Implementation

The playback system works. It places objects on the screen in the right places and can display fairly large animations at 15 frames per second, smaller ones at 30. The system is also easy to modify. The command driven design allows customizations to be made easily without having to change any more than local routines. For example, the system could be moved over to the IMI and only two routines in *View* would have to be changed.

The current version of the playback system successfully implements a scene assembler called *Prep* and a display program called *View*. Both programs were written in C on the IRIS. (*Prep* also runs on a UNIX VAX). The programs communicate through the exchange of ASCII animation files. The next two sections contain a general discussion of how these programs work.

### 4.2.2. Program Prep

*Prep* is an altered version of Crow's scene assembler (*scn\_asmblr.c*). The routines which call the high quality renderers, the routines which output written descriptions of the scenes and the routine which handles object color have been removed from the program. The commands which call these routines may be used in the scene descriptions, but they will not have any effect on the animation. Four routines have been added to the program: *hide* and *unhide*, which handle the disappearance and reappearance of objects, and *make\_frame* and *close\_frame*, which output the current scene in animation file format.

## Motion Analysis

Animations are prepared by describing the initial scene using Crow's scene description commands, putting this into an animation file using an *exit* or *close\_\_frame* command, and then describing the changes from one frame to the next, outputting the result for each frame to the animation file.

There are some important points to be made regarding the description of changes. If a particular object attribute is set in one frame and not mentioned in the next it remains unchanged. If it is desired that an object disappear, it must be explicitly removed using the *hide* command. Once hidden it will remain so until explicitly recovered by an *unhide* command. This strategy was adopted because it is more likely that an object will remain the same from one frame to the next than disappear. Finally, when an object attribute is mentioned the values given for it are treated as absolute values.

### 4.2.3. View

The current viewing program is controlled by the user. The user must make available to the program an animation file and the Crow detail file for every object in the animation. He then tells the program to load the animation file, at which point it executes all the commands in the animation file and stores the entire animation in memory.

Animations are stored as a series of object descriptions and absolute transformation matrices. There is one matrix for every object appearing in each frame. These matrices account for object position and orientation, and rotation and translation relative to the eye and view reference point. The perspective transformation is done by the IRIS with the view angle and front and back clipping planes set by commands in the animation file.

Once an animation has been read in, the user may load in other animations, enter commands which may change the current animation, or play the animation back at a specific rate and through a specific range of frames. At the moment there is no interactive control over the playback rate. The user may also, if desired, request that the program display a real-time clock and an animation clock on the screen during playback. This allows the user to closely examine the playback rate. This is important because the program may slow down and speed up depending on how many objects are actually on the screen. Some solutions for this problem are discussed in the next section.

## Motion Analysis

### 4.2.4. Necessary Extensions

There are still several areas in which the real-time playback system could be improved. The fact that the entire animation must be stored in memory presents a great limitation to the system's usefulness. There should be some kind of frame manipulator to handle animations requiring more space than is available in main memory.

There should be interactive control over the animation playback rate. Once available, this feature could be used to allow the user to give the animation system interactive feedback on the dynamic relationships between objects. This would be a useful alternative to the history list method of animation editing currently used in TEMPUS.

Interprogram communications could be improved in either of two ways. One method would be to have *View* be a slave process to *Prep*. In this way *Prep* could be running on the VAX and could send the animation file commands to *View* over Ethernet and have *View* processing them at the same time. In this configuration, the time to load an animation file would be completely eliminated, as *View* would be done as soon as *Prep* finished. Another method would be to exchange binary files instead of ASCII files. This would save time in interpreting the animation files and would also allow for more direct program structure in *View*'s main program.

### 4.3. Textual Descriptions

Since people appear to be very good at describing in language much of what they perceive visually, it is not surprising that textual descriptions of motion data are a useful method of summarizing much information. For example, the communication that someone is *walking* is easier to produce (just output the word!) than graphically display. The difficulty, however, arises in the determination of the particular motion state and not in the output of that information.

Given a static position of the human form as a stick figure, Herman [21] was able to apply pattern matching and artificial intelligence techniques to produce a textual description of body position. Outputs were of two types: *physical* and *meaning*. The physical description contains spatial relationships between body parts such as *face pointing left, knee2 partially bent, or knee2 is forward*. The meaning description is an interpretation of the (static) event depicted in the scene such as *depression, sadness, walking, or reaching-out-to-help*.

## Motion Analysis

Efforts to transform three-dimensional *motion* data into textual descriptions appears to have begun with Badler [6]. He used a vocabulary of motion *adverbials* (mostly prepositions which indicate directional movements) applied to three-dimensional trajectories of various moving objects such as a car, bouncing ball, pulley systems, and human walking motion. From these adverbials, motions that began or stopped, and other changes in the database of spatial relationships, a motion verb could be determined and uttered in a standardized natural language sentence. Badler's process was further implemented, extended, and expanded to non-rigid motions by Tsotsos [50].

While these efforts are interesting research areas, textual descriptions do not appear to be essential for the NASA effort at this time. The principal reason for this assessment is not technological feasibility, but usefulness. That is, the motion studies one would expect to analyze would probably be to assess strength or individual motion planning during zero-gravity locomotion. These tasks do not lend themselves as readily to textual summarization. The most appropriate application for textual descriptions is probably in remote personnel surveillance.

### 5. Conclusions

The conclusions we have reached in researching this subject are that human motion analysis should be performed in an interactive fashion, guided by a complete human body model, and augmented by intelligence in determining actual feature points on image sequence data.

- The data from image sources should be integrated with any other simultaneously available non-image sensor by displaying both on a suitable graphics playback device.
- The architecture of the semi-automatic image analysis system is the favorite method for expected tasks based on known costs and technological feasibility. The Galatea model provides a starting prototype for system development, while the constraint network provides the first pass at incorporating reasonable human body model intelligence in the heretofore manual digitization process.
- TEMPUS has a suitable body model for joint position determination, but would require extension with a more general kinematics system to adequately handle segment orientations. The anthropometric models in TEMPUS can be used to form the initial segment length guess needed for the analysis of figures who may not necessarily be in the current TEMPUS database.
- A real-time playback system and color coded motion parameters would form an effective tool for validating motion analyses. This system could be realized on a raster graphics workstation with real-time display and video

## Motion Analysis

overlay capabilities.

### 6. Schedule and Resources

The tasks outlined in the Conclusion could be realized over a three year period if suitable personnel were directed to its implementation. The schedule would, of course, differ if other directions were taken. In particular, the manual analysis method would take only a year, while the automatic methods could easily run to a five year project. The approximate timetable for a human motion analysis system from video or film image sequence input is given in Table 6-1.

**Table 6-1: Motion Analysis Schedule**

Time Milestone	Task (per staff member)
year 0.5	Incorporate constraint propagation into TEMPUS Build interactive data collection interface software
year 1	Build playback control system and graphical overlay. Build motion editing/filtering software.
year 2	Integrate constraints into interactive system. Compute desired motion parameters.
year 2.5	Incorporate graphical display of motion parameters. Validation, testing, and documentation.
year 3	Semi-automatic system project completion. Design better feature detectors for more automatic operation.
year 4	Implement and integrate feature detectors. Incorporate orientation constraints into model.
year 5	Nearly automatic feature detection and position analysis.

The *time milestone* is the length of time from project inception (not a duration) to the completion of the indicated *tasks*. The tasks are a summary of the work needed to fulfill the system requirements discussed in the Conclusion. Each task refers to *one* graduate research assistant. This is a half time load (20 hours/week). Thus multiple tasks for one time milestone are assumed to proceed in parallel, and a total of two individuals for five years are required. There is no guarantee that the system will be fully automatic at the end of the fifth year.

The resources required are summarized in Table 6-2. The monetary estimates are based on solely on 1985 University of Pennsylvania rates including employee benefits,

## Motion Analysis

tuition, and overhead as applicable. There is no provision for inflation; that may be projected by NASA as necessary.

**Table 6-2: Motion Analysis Resources**

2 Graduate Research Assistants for duration of project.....	\$50K/year	
Faculty supervision time (10% of academic year).....	10K/year	
Equipment:		
Raster graphics workstation with real-time capability.....	50K	\
Digitizing tablet.....	2K	} 62K
Video digitizing camera.....	5K	/
Video and graphics mixer.....	5K	/
Travel, supplies, computer, maintenance, duplicating, etc.....	40K/year	

-----

**Totals:**

Year 1:	\$152K	(includes graphics workstation and digitizer)
Year 2:	\$100K	
Year 3:	\$110K	(includes camera and video mixer)
Year 4:	\$100K	
Year 5:	\$100K	

-----

### 7. Bibliography

1. J. K. Aggarwal and W. N. Martin. Dynamic scene analysis. TR-82-3, The University of Texas at Austin, Laboratory for Image and Signal Analysis, September, 1982.
2. J. K. Aggarwal. Three-dimensional description of objects and dynamic scene analysis. TR-83-1-20, The University of Texas at Austin, Laboratory for Image and Signal Analysis, January, 1983.
3. Koichiro Akita. Analysis of body motion image sequences. Proceedings of the 6th International Conference on Pattern Recognition, October, 1982, pp. 320-327.
4. Minoru Asada, Masahiko Yachida and Saburo Tsuji. Understanding of three-dimensional motions in blocks world. Control Engineering 81-01, Osaka University, April, 1981.
5. Minoru Asada and Saburo Tsuji. "Representation of three-dimensional motion in dynamic scenes". *Computer Vision, Graphics, and Image Processing* 21, 1 (January 1983), 118-144.
6. Norman I. Badler. *Temporal scene analysis: Conceptual descriptions of object movements*. Ph.D. Th., University of Toronto, Dept. of Computer Science, 1975.
7. Norman I. Badler, Jon Korein, Craig Meyer, Kamran Manoochehri, Jane Rovins, Jeffrey Beale, Brian Barr. System Integration Report. Dept. of Computer and Information Science, University of Pennsylvania, Philadelphia, PA, 1985. (For NAS9-16634).
8. P. Bapu, S. Evans, P. Kitka, M. Korna, and J. McDaniel. User's guide for COMBIMAN programs. Univ. of Dayton Research Institute, January, 1981. U.S.A.F. Report No. AFAMRL-TR-80-91.
9. S. T. Barnard and W. B. Thompson. "Disparity Analysis of Images". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2, 4 (July 1980), 333-340.
10. W. Brugger and Morris Milner. "Computer-aided tracking of body motions using a CCD-image sensor". *Medical and Biological Engineering and Computing* (1977).
11. T. Calvert, J. Chapman and A. Patla. "The integration of subjective and objective data in the animation of human movement". *Computer Graphics* 14, 3 (July 1980), 198-203.
12. Zen Chen and Hsi-Jian Lee. Recovery of three-dimensional human motion from an image sequence. The Institute of Computer Engineering, January, 1985. National Chiao-Tung University, Hsinchu, Taiwan, R.O.C..
13. Frank C. Crow. "A more flexible image generation environment". *Computer Graphics* 16, 3 (July 1982), 9-18.
14. J. E. Cutting. "A Program to Generate Synthetic Walkers as Dynamic Point-Light Displays". *Behaviour Research Methods and Instrumentation* 10, 1 (1978), 91-94.
15. Tom DeWitt and Phil Edelstein. "Pantomation". *SIGGRAPH Video Review* 2 (September 1981). Electronic Body Arts, Inc., Albany, New York.



## Motion Analysis

16. Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, 1973.
17. J. Q. Fang and T. S. Huang. "Some experiments on estimating the 3-D motion parameters of a rigid body from two consecutive image frames". *IEEE Trans. on Pattern Analysis and Machine Intelligence* 6 (1984), 545-554.
18. Carol M. Ginsberg and Delle Maxwell. "Graphical Marionette": A Modern Day Pinocchio. Workshop on Motion: Representation and Perception, Toronto, Canada, April, 1983.
19. Michael Girard and A. A. Maciejewski. "Computational modeling for the computer animation of legged figures". *Computer Graphics* 19 (1985). to appear.
20. W. E. L. Grimson. *From Images to Surfaces: A Computational Study of the Human Early Visual System*. MIT Press, Cambridge, MA, 1981.
21. Martin Herman. A computer program for generating the semantics of human stick figure. Computer Science TR-729, University of Maryland, January, 1979.
22. Michael Jenkin and John K. Tsotsos. "Applying temporal constraints to the dynamic stereo problem". *Computer Vision, Graphics and Image Processing* (1986). (to appear).
23. JML Research, Inc. IMP-84. Madison, WI.
24. G. Johansson. "Spatial-temporal Differentiation and Integration in Visual Motion Perception". *Psychology Research* 38 (1976), 379-383.
25. T. Kasvand and Morris Milner. "Pattern recognition applied to measurement of human limb positions during movement". *Journal of Cybernetics* 2, 1 (1972), 66-78.
26. Chattecx Corp. Kin-Com. Chattanooga, TN.
27. E. Kingsley, N. Schofield, and K. Case. "SAMMIE-a computer aid for man-machine modeling". *Computer Graphics* 15, 3 (August 1981), 163-169.
28. James U. Korein. *A geometric investigation of reach*. MIT Press, Cambridge, MA, 1985.
29. Hsi-Jian Lee and Zen Chen. "Determination of 3D human body postures from a single view". *Computer Vision, Graphics and Image Processing* 30, 2 (1985), 148-168.
30. Steven A. MacKay, Richard E. Sayre, and Michael J. Potel. "3D Galatea: Entry of three-dimensional moving points from multiple perspective views". *Computer Graphics* 16, 3 (1982), 213-222.
31. D. Marr and T. Poggio. "A computational theory of human stereo vision". *Proceedings of the Royal Society of London B*, 204 (1979), 301-328.
32. David Marr and Lucia Vaina. Representation and recognition of the movements of shapes. Computer Science TR-597, Massachusetts Institute of technology, October, 1980.
33. W. N. Martin and J. K. Aggarwal. "Dynamic Scene Analysis: A Survey". *Computer Graphics and Image Processing* 7, 3 (June 1978), 356-374.

## Motion Analysis

34. A. Z. Meiri. "On Monocular Perception of 3D Moving Objects". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2, 6 (November 1980), 582-583.
35. Pietro Morasso and Vincenzo Tagliasco. Robust three-dimensional localization of points and lines by means of multiple images. Computer Science TR-161, University of Genoa, 1984.
36. J. O'Rourke. *Image Analysis of Human Motion*. Ph.D. Th., University of Pennsylvania, Moore School of Electrical Engineering, Philadelphia, PA, 1978.
37. Joseph O'Rourke and Norman I. Badler. "Model-based image analysis of human motion using constraint propagation". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2 (November 1980), 522-536.
38. Joseph O'Rourke. Dynamically quantized spaces applied to motion analysis. JHU-EE 81-1, The Johns Hopkins University, January, 1981.
39. N. Orlandea and D. A. Calahan. A sparsity-oriented approach to the design of mechanical systems. In *Problem Analysis in Science and Engineering*, Academic Press, New York, 1977.
40. Burton Paul and Ronald Schaffa. *DYSPAM User's Manual*. Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania.
41. John P. Paul. "Kinetic measurement systems and biomechanical data". *Journal of Human Movement Studies* 4, 4 (1977), 179-190.
42. Richard P. Paul. *Robot Manipulators: Mathematics, Programming, and Control*. MIT Press, Cambridge, MA, 1981.
43. Howard Poizner, Ursula Bellugi, and Venita Lutes-Driscoll. "Perception of American sign language in dynamic point-light displays". *Journal of Experimental Psychology* 7, 2 (1981), 430-440.
44. Michael J. Potel, Richard E. Sayre, and Steven A. MacKay. "Graphics input tools for interactive motion analysis". *Computer Vision, Graphics and Image Processing* 14, 1 (1980), 1-23.
45. Richard F. Rashid. "Towards a system for the interpretation of moving light displays". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2 (November 1980), 574-581.
46. D. Rogers and J. Adams. *Mathematical Elements for Computer Graphics*. McGraw-Hill, New York, 1976.
47. Glen Speckert. A Computerized Look at Cat Locomotion, or, One Way to Scan a Cat. A. I. Memo 374, M.I.T. Artificial Intelligence Laboratory, July, 1976.
48. Scott N. Steketee and Norman I. Badler. "Parametric keyframe interpolation incorporating kinetic adjustment and phrasing control". *Computer Graphics* 19 (1985). to appear.
49. V. Tartter and K. C. Knowlton. "Perception of Sign Language from an Array of 27 Moving Spots". *Nature* 289 (February 1981), 676.

## Motion Analysis

50. John K. Tsotsos. A framework for visual motion understanding. CSRG 114, Dept. of Computer Science, University of Toronto, 1980.
51. Saburo Tsuji, Michiharu Osada and Masahiko Yachida. "Tracking and segmentation of moving objects in dynamic line images". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2, 6 (November 1980), 516-522.
52. Shimon Ullman. "The Interpretation of Structure from Motion". *Proceedings of the Royal Society of London B*, 203 (1979), 405-426.
53. Shimon Ullman. Maximizing Rigidity: The incremental recovery of 3-D structure from rigid and rubbery motion. AI-TR-721, Massachusetts Institute of Technology, June, 1983.
54. Harald G. Wallbott. "The measurement of human expression". *Aspects of Nonverbal Communication* (1980), 203-228.
55. Jon A. Webb and J. K. Aggarwal. "Structure from motion of rigid and jointed objects". *Artificial Intelligence* 19 (September 1982), 107-130.
56. Jon A. Webb and J. K. Aggarwal. "Shape and correspondence". *Computer Vision, Graphics, and Image Processing* 21, 1 (January 1983), 145-160.
57. Herman J. Woltring. "Planar control in multi-camera calibration for 3-D gait studies". *Journal of Biomechanics* 13, 1 (1980), 39-48.
58. Masahiko Yachida, Motozo Ikeda and Saburo Tsuji. "A Plan-guided analysis of cineangiograms for measurement of dynamic behavior of heart wall". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2, 6 (November 1980), 537-543.
59. B. L. Yen and T. S. Huang. "Determining 3-D motion and structure of a rigid body using the spherical projection". *Computer Vision, Graphics, and Image Processing* 21, 1 (January 1983), 21-32.